# Chapter 16
## Configure Frame Relay

The Frame Relay protocol allows network designers to reduce costs by using shared facilities that are managed by a Frame Relay service provider. Users pay fixed charges for the local connections from each site in the Frame Relay network to the first Point of Presence (POP) in which the provider maintains a Frame Relay switch. The portion of the network between the end point switches is shared by all the customers of the service provider, and individual Data Link Connection Identifiers (DLCIs) are assigned to ensure each customer receives only their own traffic.

Users contract with their providers for a specific minimum portion of the shared bandwidth Committed Information Rate (CIR) and for a maximum allowable peak rate, Burst Information Rate (BIR). Depending on the terms of the contract, traffic exceeding the CIR can be marked as eligible for discard, in the event of network congestion, or a best effort term can apply up to the BIR rate.

Frame Relay does not require private and permanently connected Wide Area Network facilities, unlike some older WAN protocols.

Frame Relay was developed as a replacement for the older and much slower X.25 protocol. It scales to much higher data rates because it does not require explicit acknowledgment of each frame of data.

You can configure the Frame Relay protocol on SONET/SDH, E1/E3 and T1/T3 physical router interfaces, as well as the Channelized DS-3 and Channelized OC-12 interfaces.

This chapter discusses the following Frame Relay properties that you can configure:

Configure Interface Encapsulation on page 252

Configure the Media MTU on page 253

Set the Protocol MTU on page 253

Configure Keepalives on page 253

Configure Inverse Frame Relay ARP on page 255

Configure the Router as a DCE on page 255

Configure Frame Relay DLCIs on page 255

# Configure Interface Encapsulation

Point-to-Point Protocol (PPP) encapsulation is the default encapsulation type for physical interfaces. You need not configure encapsulation for any physical interfaces that support PPP encapsulation. If you do not configure encapsulation, PPP is used by default. For physical interfaces that do not support PPP encapsulation, you must configure an encapsulation to use for packets transmitted on the interface. You can optionally configure an encapsulation on a logical interface, which is the encapsulation used within certain packet types.

## Configure the Encapsulation on a Physical Interface

For Frame Relay interfaces, you configure Frame Relay encapsulation on the physical interface. This encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. SONET and T3 interfaces can use Frame Relay encapsulation.

To configure Frame Relay encapsulation on a physical interface, include the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level, specifying the frame-relay, frame-relay-ccc, or frame-relay-tcc option:

```
[edit interfaces interface-name]
encapsulation (frame-relay | frame-relay-ccc | frame-relay-tcc);
```

When you configure a multipoint encapsulation (such as Frame Relay), the physical interface can have multiple logical units, and the units can be either point to point or multipoint.

### Example: Configure the Encapsulation on a Physical Interface

Configure Frame Relay encapsulation on a SONET interface. The second and third family statements allow IS-IS and MPLS to run on the interface.

```
[edit interfaces]
so-7/0/0 {
    encapsulation frame-relay;
    unit 0 {
        point-to-point;
        family inet {
            address 192.168.1.113/32 {
                destination 192.168.1.114;
            }
        }
        family iso;
        family mpls;
    }
}
```

## Configure the Encapsulation on a Logical Interface

Generally, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for Frame Relay encapsulation, you can also configure the encapsulation type that is used inside the Frame Relay packet itself. To do this, include the encapsulation statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level, specifying the frame-relay-ccc or frame-relay-tcc option:

```
[edit interfaces interface-name unit logical-unit-number]
encapsulation (frame-relay-ccc | frame-relay-tcc);
```

## Configure the Media MTU

For Frame Relay interfaces, the default media MTU is 4482 bytes. (For a complete list of MTU values, see Table 3 on page 41 and Table 6 on page 43.)

To modify the default media MTU size for a physical interface, include the mtu statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
mtu bytes;
```

If you change the size of the media MTU, you must ensure that the size is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. You configure the protocol MTU by including the mtu statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level, as discussed in "Set the Protocol MTU" on page 253.

## Set the Protocol MTU

For each interface, you can configure an interface-specific MTU by including the mtu statement at the [edit interfaces interface *interface-name*] hierarchy level. If you need to modify this MTU for a particular protocol family, include the mtu statement at the [edit interfaces interface *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
mtu mtu;
```

For Frame Relay encapsulation, the default protocol MTU is 4470 bytes.

If you increase the size of the protocol MTU, you must ensure that the size of the media MTU is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. (You configure the media MTU by including the mtu statement at the [edit interfaces *interface-name*] hierarchy level, as discussed in "Configure the Media MTU" on page 253.)

When the family is mpls, the default protocol MTU is 1488 bytes. MPLS packets are 1500 bytes and have 4 to 12 bytes of overhead.

## Configure Keepalives

By default, physical interfaces configured with Cisco HDLC or PPP encapsulation send keepalive packets at 10-second intervals. The Frame Relay term for keepalives is Local Management Interface (LMI) packets; note that the JUNOS software supports both ANSI T1.617 Annex D LMIs and ITU Q933 Annex A LMIs.

To disable the sending of keepalives on a physical interface, include the no-keepalives statement at the [edit interfaces interface *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
no-keepalives;
```

For back-to-back Frame Relay connections, either disable the sending of keepalives on both sides of the connection, or configure one side of the connection as a DTE (the default JUNOS configuration) and the other as a DCE.

If keepalives are enabled, the number of possible DLCI configurations on a multipoint or multicast connection is limited by the MTU size selected for the interface. To calculate the available DLCIs, use the formula (MTU – 12) /5. To increase the number of possible DLCIs, disable keepalives.

## Configure Tunable Keepalives for Frame Relay LMI

On interfaces configured with Frame Relay connections, you can tune the keepalive settings by using the lmi statement. A Frame Relay interface can be either data circuit-terminating equipment (DCE) or data terminal equipment (DTE) (the default JUNOS configuration). DTE acts as a master, requesting status from the DCE part of the link.

By default, the JUNOS software uses ANSI T1.617 Annex D LMIs. To change to ITU Q933 Annex A LMIs, include the lmi-type itu statement at the [edit interfaces *interface-name* lmi] hierarchy level:

```
[edit interfaces interface-name lmi]
lmi-type itu;
```

To configure Frame Relay keepalive parameters, include the lmi statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
lmi {
   lmi-type (ansi | itu);
   n391dte number;
   n392dce number;
   n392dte number;
   n393dce number;
   n393dte number;
   t391dte seconds;
   t392dce seconds;
}
```

You can set the following parameters:

n391dte—DTE full status polling interval. The DTE sends a status inquiry to the DCE at the interval specified by t391dte. n391dte specifies the frequency at which these inquiries expect a full status report; for example, a n391dte value of 10 would specify a full status report in response to every tenth inquiry. The intermediate inquiries ask for a keepalive exchange only. The range is 1 through 255, with a default value of 6.

n392dce—DCE error threshold. The number of errors required to bring down the link, within the event-count specified by n393dce. The range is 1 through 10, with a default value of 3.

n392dte—DTE error threshold. The number of errors required to bring down the link, within the event-count specified by n393dte. The range is 1 through 10, with a default value of 3.

n393dce—DCE monitored event-count. The range is 1 through 10, with a default value of 4.

n393dte—DTE monitored event-count. The range is 1 through 10, with a default value of 4.

t391dte—DTE keepalive timer. Period at which the DTE sends out a keepalive response request to the DCE and updates status depending on the DTE error threshold value. The range is 5 through 30 seconds, with a default value of 10 seconds.

t392dce—DCE keepalive timer. Period at which the DCE checks for keepalive responses from the DTE and updates status depending on the DCE error threshold value. The range is 5 through 30 seconds, with a default value of 15 seconds.

## Configure Inverse Frame Relay ARP

Frame Relay interfaces support inverse Frame Relay ARP, as described in RFC 2390. When inverse Frame Relay ARP is enabled, the router responds to received inverse Frame Relay ARP requests by providing IP address information to the requesting router on the other end of the Frame PVC (permanent virtual circuit).

The router does not initiate inverse Frame Relay ARP requests.

By default, inverse Frame Relay ARP is disabled. To configure a router to respond to inverse Frame Relay ARP requests, include the inverse-arp statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* multipoint-destination *destination*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
inverse-arp;
```

You must configure Frame Relay encapsulation on the logical interface to support inverse ARP. For more information, see “Configure Interface Encapsulation” on page 252.

## Configure the Router as a DCE

By default, when you configure an interface with Frame Relay encapsulation, the router is assumed to be data terminal equipment (DTE). That is, the router is assumed to be at a terminal point on the network. To configure the router to be data circuit-terminating equipment (DCE), include the dce statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
dce;
```

When you configure the router to be a DCE, keepalives are disabled by default.

For back-to-back Frame Relay connections, either disable the sending of keepalives on both sides of the connection, or configure one side of the connection as a DTE (the default JUNOS configuration) and the other as a DCE.

## Configure Frame Relay DLCIs

When you are using Frame Relay encapsulation on an interface, each logical interface corresponds to one or more permanent virtual circuits (PVCs) or switched virtual circuits (SVCs). For each PVC or SVC, you must configure one data-link connection identifier (DLCI).

A Frame Relay interface can be a point-to-point interface or a point-to-multipoint (also called a multipoint nonbroadcast multiaccess [NBMA]) connection.

To configure Frame Relay DLCIs, you can do the following:

Configure a Point-to-Point Frame Relay Connection on page 256

Configure a Point-to-Multipoint Frame Relay Connection on page 256

Configure a Multicast-Capable Frame Relay Connection on page 257

## *Configure a Point-to-Point Frame Relay Connection*

To configure a point-to-point Frame Relay connection, include the dlci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
dlci dlci-identifier;
```

*dlci-identifier* is the DLCI identifier, which is a number from 1 through 1022. Numbers 1 through 15 are reserved. A point-to-point interface can have one DLCI.

> **Note**
>
> Channelized DS-3 and Channelized STM-1 to E1 interfaces can support a maximum of 64 DLCIs per channel, for a total of 1792 DLCIs per DS-3 interface or 4032 DLCIs per STM-1 interface. For more information, see “Configure Channelized DS-3 to DS-1 Interfaces” on page 197 or “Configure Channelized STM-1 Interfaces” on page 206.

When you are configuring point-to-point connections, the MTU sizes on both sides of the connection must be the same.

## *Configure a Point-to-Multipoint Frame Relay Connection*

To configure a point-to-multipoint Frame Relay connection (also called a multipoint NBMA connection), include the multipoint-destination statement within the address statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
address address {
    multipoint-destination destination-address dlci dlci-identifier;
}
```

*address* is the interface’s address.

For each destination, include one multipoint-destination statement. *destination-address* is the address of the remote side of the connection, and *dlci-identifier* is the DLCI identifier for the connection.

When you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

If keepalives are enabled, causing the interface to send LMI messages during idle times, the number of possible DLCI configurations is limited by the MTU selected for the interface. For more information, see “Configure Keepalives” on page 253.

## Configure a Multicast-Capable Frame Relay Connection

By default, Frame Relay connections assume unicast traffic. If your Frame Relay switch performs multicast replication, you can configure the connection to support multicast traffic by including the multicast-dlci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
multicast-dlci dlci-identifier;
```

*dlci-identifier* is the DLCI identifier, which is a number from 1 through 1022 that defines the Frame Relay DLCI over which the switch is expecting to receive multicast packets for replication.

You can configure multicast support only on point-to-multipoint Frame Relay connections.

If keepalives are enabled, causing the interface to send LMI messages during idle times, the number of possible DLCI configurations is limited by the MTU selected for the interface. For more information, see “Configure Keepalives” on page 253.